1) define a functional interface First with an abstract method void disp1() , default method void disp2() and a static method void disp3()

inside main function using lambda expression invoke disp1() and disp2() methods.

also invoke disp3() method inside main.

interface First{

void disp1() ;

default void disp2() {

System.***out***.println("default void disp2() ");

}

static void disp3() {

System.***out***.println("Static void disp3()");

}

}

public class Demo{

public static void main(String[] args) {

First f = () -> { System.***out***.println("void disp1() ");};

f.disp1();

f.disp2();

First.*disp3*();

}

}

2) define a functional interface Second with an abstract method void disp4().

define a class Demo with main function.

inside main function create two implementations of Second using lambda expression and display their names.

interface Second{

void disp4() ;

}

public class Demo{

public static void main(String[] args) {

Second s = () ->{System.***out***.println("disp4 first");};

s.disp4();

Second s2 = () -> {System.***out***.println("disp4 2nd");};

s2.disp4();

}

}

3) define a functional interface MyInterface with an abstract method void fun()

define a class Demo with two functions static void perform and main.

from main function pass lambda expression, collect it in perform method and invoke fun() function.

interface MyInterface {

void fun() ;

}

public class Demo{

static void perform (MyInterface ref) {

ref.fun();

}

public static void main(String[] args) {

*perform*(() -> {System.***out***.println("fun()");});

}

}

4) define a functional interface Calculator with an abstract method int multiply(int a,int b)

define a class CalcDemo with main method.

inside main using lambda create an implementation of Calculator , invoke multiply() method and display its result.

interface Calculator {

int multiply(int a,int b) ;

}

public class CalcDemo{

public static void main(String[] args) {

Calculator c = (a,b) -> { return a\*b; };

System.***out***.println(c.multiply(10, 2));

}

}

5) define a class Display with void disp method. inside this method display values from 1 to 10.

create an implementation of Runnable using lambda expression which will invoke disp() method of Display class.

create 2 threads and pass above created Runnable implementation to these threads.

make sure while one thread is executing other thread will not interfere.

public class Display{

synchronized void disp () {

for(int i=0;i<10;i++) {

System.***out***.print((i+1) + " ");

}

System.***out***.println();

}

public static void main(String[] args) {

Display d = new Display();

Runnable ref = () -> {d.disp();};

Thread t1 = new Thread(ref);

Thread t2 = new Thread(ref);

t1.start();

t2.start();

}

}